

**TECHNICKÁ UNIVERZITA V LIBERCI**

Fakulta mechatroniky, Informatiky a mezioborových studií

Studijní program: B2612 Elektrotechnika a informatika

Studijní obor: 1802R022 Informatika a logistika

## **Webová aplikace pro správu softwarových licencí pro organizace**

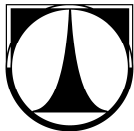
## **Web Application for a Software License Management of Small Organizations**

Bakalářská práce

**Autor:** Tomáš Buček

**Vedoucí práce:** Ing. Petr Kretschmer

**Datum:** 21.5.2010



## **PODĚKOVÁNÍ**

Děkuji Ing. Petru Kretschmerovi za hodnotné rady a odborné vedení během mé práce. Děkuji svým kolegům za pochopení a své ženě za trpělivost.



## **PROHLÁŠENÍ**

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 - Školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

V Sokolově 21.5.2010

**Tomáš Buček**



Název práce: Webová aplikace pro správu softwarových licencí pro organizace

Autor: Tomáš Buček

Fakulta: Mechatroniky, informatiky a mezioborových studií

Vedoucí bakalářské práce: Ing. Petr Kretschmer, Ústav Nových technologií a aplikované informatiky, Technická univerzita v Liberci, Studentská 2, 461 17 Liberec 1

e-mail vedoucího: petr.kretschmer@tul.cz

Abstrakt: Aplikace je určena pro evidenci programového vybavení a počítačových sestav. Umožňuje správu programových aplikací, zobrazení počtu dostupných licencí, souhrnné zobrazení programů na určitém počítači, tisk sestav do PDF. Nabízí 3 úrovně uživatelských práv. Napsána je v jazyce PHP, využívá databázi MSSQL Express. Pro tisk sestav je použita volně dostupná knihovna FPDF. Popis aplikace je doplněn obrázky aplikace, je popsán PHP skript a základní příkazy FPDF.

Klíčová slova: Evidence programů, databáze, PHP, SQL, tisk do PDF

---

Title of the work: Web Application for a Software License Management of Small Organizations

Author: Tomáš Buček

Faculty: of Mechatronics, Informatics and Interdisciplinary Studies

Tutor of the Bachelor work: Ing. Petr Kretschmer, Ústav Nových technologií a aplikované informatiky, Technical university in Liberec, Studentská 2, 461 17 Liberec 1

E-mail of the Tutor: petr.kretschmer@tul.cz

Abstract:

Application is used for stocking of system software and computing equipment. It is provide for administration of computer application, displaying numbers of accessible licenses, conjunctive software display on specific computer, printing outputs in PDF. It offers 3 levels of user's authorization. It is written in PHP language and uses database MSSQL Express. As for printing of outputs is used FPDF free library. Description of application also consist of pictures of this application, there are also described PHP script and basic directions of FPDF.

Key words: stocking of system software, database, PHP, SQL, printing into PDF



## Obsah

PROHLÁŠENÍ .....	3
1. Zadání.....	6
2. Návrh řešení a použité technologie .....	6
3. Databáze MSSQL, resp. její Express Edition Verze.....	7
4. Jazyk PHP a jeho možnosti použití .....	15
5. Tisková knihovna FPDF.....	20
Základní příkazy FPDF .....	21
6. Instalace aplikace .....	22
7. Vlastní návrh aplikace .....	24
7.1 Přihlašovací obrazovka .....	24
7.2 Evidence přihlášení uživatelů .....	24
7.3 Úvodní obrazovka .....	25
7.4 Evidence programových aplikací .....	26
7.5 Evidenční list programové aplikace .....	27
7.6 Evidenční list update/upgrade programové aplikace.....	27
7.7 Tisk stránky do PDF.....	28
7.8 Update a upgrade .....	28
7.9 Evidence uživatelů .....	29
7.10 Evidence PC.....	30
7.11 Přidání / editace PC .....	30
7.12 Přiřazení PC k software .....	31
7.13 Tisková sestava .....	31
8. Přílohy .....	33
8.1 Struktura databáze MSSQL.....	33
8.2 Schéma databáze.....	34
8.3 Výpis skriptů pro vytvoření tabulek:.....	34
9. Ukázkový popis skriptů .....	37
9.1 Skript pro zobrazení aplikace.....	37
9.2 Skript pro tisk do PDF .....	41
10. Závěr.....	45
Použité zdroje:.....	46



## 1. Zadání

Vytvoření aplikace v jazyce PHP na tvorbu dynamických webových stránek. Cílem projektu je vytvořit rozhraní pro vkládání dat, jejich editaci a prezentaci. Řízení přístupu k funkcím aplikace bude založeno na různých uživatelských právech.

## 2. Návrh řešení a použité technologie

Úkolem projektu bylo vytvoření jednoduché aplikace v jazyce PHP na tvorbu dynamických webových stránek. Po konzultaci s vedoucím práce Ing. Kretschmerem, jsem zvolil problematiku evidence programového vybavení, PC a vazby mezi programy a PC.

Aplikace byla původně napsána pouze pro evidenci programového vybavení, poté byla rozšířena o evidenci PC.

Aplikace byla vyvíjena pro použití na mém pracovišti (Krajský úřad Karlovarského kraje) nebo pro potřeby TUL, jako evidenční systém programového vybavení s funkcemi jako je hlídání počtu licencí, upgradů a nových verzí, tvorba tiskových sestav pro určitý počítač (a tím i určitého pracovníka), atd.

Je umožněno vkládání, editace a mazání programů, vkládání, editace a mazání upgradů (nových verzí) k příslušnému programu, vkládání, editace a mazání počítačů, přidružování programu k PC, práce s uživateli (podle práva přístupu jsou některé funkce nedostupné), jednoduchý export do PDF.

Aplikace využívá databázi MSSQL 2005 od firmy Microsoft, konkrétně její verzi SQL Server 2005 Express Edition (<http://www.microsoft.com/sqlserver/2005/en/us/express.aspx>) pomocí ODBC connectoru a je vytvořena v jazyce PHP, verze 5 ([www.php.net](http://www.php.net)). Do databáze přistupuje pod jedním uživatelským jménem, tento uživatel je zároveň vlastník databáze.

Pro export do PDF byla použita volně dostupná knihovna FPDF ([www.fpdf.org](http://www.fpdf.org)). Bližšímu popisu těchto technologií a produktů věnuji další části textu.



### 3. Databáze MSSQL, resp. její Express Edition Verze

Zkratka *SQL* znamená *Structured Query Language* a jedná se o neprocedurální jazyk. (Procedurální jazyk říká, *jak chceme konkrétní věc provést*, neprocedurální jazyk naopak říká, *co chceme provést*.)

První prototypová implementace tohoto jazyka, který se až do konce 70. let nazýval *Sequel*, vznikla roku 1974 v laboratořích společnosti *IBM* a byla použita ve firemním systému s názvem *R*.

Cílem bylo (a stále je) poskytnout vývojářům standardní metodu přístupu k datům uloženým v databázovém systému, která by byla nezávislá na dalších použitých vývojových nástrojích. Na jednu stranu se jedná o jazyk velmi mocný a zároveň poměrně jednoduchý (stačí si do angličtiny přeložit, co vlastně chceme, a samotný příkaz bude vypadat velmi podobně), současně však není samostatně dostačující k tvorbě kvalitní aplikace (SQL například nemá implementovány příkazy IO, tzn. pro uživatelský vstup a výstup). [1]

Jazyk *SQL* také není možné chápat striktně jako dotazovací, jak by se mohlo dle slova *Query* (dotaz) v názvu zdát - samotný *SQL* se skládá ze čtyř částí.

- DDL - příkazy pro manipulaci s daty (SELECT, INSERT, UPDATE, DELETE, ...)
- DML - příkazy pro definici dat (CREATE, ALTER, DROP, ...)
- DCL - příkazy pro řízení přístupových práv (GRANT, REVOKE)
- TCC - příkazy pro řízení transakcí (START TRANSACTION, COMMIT, ROLLBACK)

Pomocí příkazů *DDL* se definují vlastní databázové struktury, ať již ve tvaru fyzického uložení dat (jako jsou tabulky či indexy), či abstraktní, logické (například pohledy, tzv. VIEW tabulky). VIEW tabulky se tváří navenek jako normální tabulky, ale uvnitř jsou dynamicky tvořeny dotazy na jednu či více tabulek.

Naproti tomu pomocí příkazů *DML* s daty opravdu "manipulujeme", tedy vkládáme je (INSERT), provádíme nad nimi dotazy (SELECT) či je upravujeme (UPDATE, DELETE). Stranou v mnoha systémech nezůstávají ani prostředky pro řízení transakčního zpracování.

Přestože asi většina z nás má pojem *SQL* spojen s některým relačním *SŘBD* (systém řízení báze dat), není nikde stanoveno, že by nebyl použitelný v systémech založených na jiném modelu, nebo že všechny relační databáze musí *SQL* podporovat. Nikde také není řečeno,



že se musí jednat o jazyk interpretační či vyžadující kompilaci. Možné jsou všechny tři varianty - čistá interpretace, překlad do pre-kódu i překlad do samostatně spustitelného kódu. Přitom, jak už to tak bývá, každá varianta má své výhody a nevýhody. Například, pokud se jedná o program zkompilovaný do samostatně spustitelného kódu, je nutno zajistit jeho rekompilaci, pokud dojde k nějaké změně v použitých databázových strukturách. [1]

Stejně jako v případě celé řady dalších programovacích jazyků se i *SQL* může pochlubit poměrně živelným vývojem. Na trhu se v první polovině 80. let objevily asi čtyři desítky komerčních *SQL* systémů. Situace začala být poměrně vážná, a tak se přikročilo k prvnímu normalizačnímu kroku. [1]

Organizace *ANSI* přijala roku 1986 jako standard variantu společnosti *IBM* a ve svém důsledku se jednalo o jakýsi průnik existujícími implementacemi. Tento standard bývá označován jako *ANSI SQL* či jako *SQL-86*. O rok později byl přijat také v *ISO*. Roku 1989 pak následoval další standard, který umožnil definovat integritní omezení. Vývoj šel nezadržitelně kupředu a roku 1992 dochází k přijetí standardu *SQL-92 (nebo též nazývaný SQL2)*, o kterém se již v době vzniku vědělo, že se nejedná o konečnou podobu standardu *SQL*. V roce 1999 přichází verze *SQL:1999 (SQL3)*, která přidala možnost používání rekurzivních dotazů, non-skalárních typů, a některé objektově orientované rysy. V roce 2003 přichází verze *SQL:2003*, která nabízí podporu XML, normalizovaných sekvencí a sloupce s automaticky vkládanými hodnotami. Další verzí je *SQL:2006*, která definuje způsoby, jak importovat a ukládat XML data v *SQL* databázi, jak s těmito daty manipulovat a jak je publikovat. Publikovat se dají XML i standardní *SQL* data ve formě XML. Navíc tato verze umožňuje aplikacím integrovat jejich *SQL* kód použitím *XQUERY*, což je XML Query jazyk publikovaný konsorciem W3C, pro přístup k *SQL* a XML dokumentům. Poslední verzí je *SQL:2008* která zavádí jen menší změny (přidání vlastnosti *INSTEAD OF* triggerům, přidává stav *TRUNCATE*, atd.)

## **Datová platforma serveru SQL Server**

*SQL Server* je všestranné, integrované a komplexní řešení firmy *MICROSOFT* pro data, které přispívá ke zvýšení výkonnosti uživatelů v celé organizaci poskytováním zabezpečenější, spolehlivější a produktivnější platformy pro podniková data a aplikace BI.

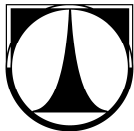




Obrázek 1: Rozložení datové platformy serveru SQL Server 2003 ([www.microsoft.cz](http://www.microsoft.cz))

***Datová platforma serveru SQL Server zahrnuje následující nástroje:***

- **Relační databáze** – zabezpečený a spolehlivý vysoce dostupný relační databázový stroj se zvýšeným výkonem a podporou strukturovaných a nestruturovaných dat (XML).
- **Služba Replication Services** - replikace dat pro aplikace zpracovávající distribuovaná či mobilní data
- **Služba Notification Services** - pokročilé funkce zasílání upozornění pro vývoj a nasazení škálovatelných aplikací
- **Služba Integration Services** - funkce extrakce, transformace a načítání dat (ETL) pro datové sklady a integraci dat v celém podniku.
- **Služba Analysis Services** - funkce OLAP (Online Analytical Processing) pro rychlou a pokročilou analýzu velkých a složitých datových sad s využitím vícedimenzionálních úložišť.
- **Služba Reporting Services** - komplexní řešení pro vytváření, správu a zasílání tradičních papírových i interaktivních webových sestav.
- **Nástroje pro správu** - SQL Server zahrnuje integrované nástroje pro pokročilou správu a ladění databází a umožňuje úzkou integraci s nástroji, jako jsou například Microsoft Operations Manager (MOM) a Microsoft Systems Management Server (SMS).



- **Nástroje pro vývojáře** - SQL Server nabízí integrované nástroje pro vývojáře určené pro databázový stroj, extrakci, transformaci a načítání dat (ETL), dolování dat, funkce OLAP a vytváření sestav, které jsou úzce integrovány se sadou Microsoft Visual Studio a poskytují komplexní funkce pro vývoj aplikací.
- **Analytické služby** - SQL Server 2008 usnadňuje a umožňuje organizacím pomocí známých nástrojů vytvářet komplexní podniková analytická řešení zajišťující srozumitelný přehled.
- **Dolování dat** - SQL Server 2008 podporuje informovaná rozhodnutí díky prediktivní analýze prostřednictvím uceleného a intuitivního dolování dat, které je jednoduše integrováno v rámci platformy Microsoft Business Intelligence a lze je rozšířit na podnikové aplikace.
- **Vysoká dostupnost (Always On)** - Technologie SQL Server 2008 Always On poskytují celou řadu možností pro minimalizaci prostojů a zachování odpovídajících úrovní dostupnosti aplikací.
- **Integrační služby** - SQL Server 2008 nabízí škálovatelnou platformu pro integraci podnikových dat s výjimečnými funkcemi pro extrahování, transformaci a načítání dat (ETL) a možnostmi integrace, která organizacím usnadňuje správu dat z celé řady zdrojů.
- **Spravovatelnost** - Microsoft SQL Server 2008 obsahuje systém zásad pro správu instancí systému SQL Server a také nástroje pro monitorování výkonu, odstraňování potíží a ladění, které správcům umožňují efektivnější správu databází a uvedených instancí.
- **Výkon a škálovatelnost** - SQL Server 2008 je vybaven mnoha technologiemi pro škálování jednotlivých serverů v rámci systému nebo škálování velmi objemných databází mezi systémy. Zároveň také nabízí nástroje zajišťující optimalizaci výkonu.
- **Programovatelnost** - SQL Server 2008 umožňuje vývojářům s pomocí rozhraní .NET Framework a sady Visual Studio Team System vytvářet výkonné databázové aplikace nové generace.
- **Služby zpráv** - Služba SQL Server 2008 Reporting Services nabízí komplexní serverovou platformu navrženou pro podporu nejrůznějších potřeb generování sestav. Umožňuje organizacím získat relevantní informace tam, kde jsou v podniku potřebné.



- **Zabezpečení** - Microsoft SQL Server 2008 nabízí vylepšené funkce zabezpečení, které přináší efektivní správu konfigurace funkcí zabezpečení, silné ověřování a řízení přístupu, výkonné možnosti šifrování a správy klíčů a zdokonalené auditování.
- **Prostorová data** - SQL Server 2008 poskytuje komplexní podporu prostorových dat, která organizacím umožňuje snadno zpracovávat, používat a rozšiřovat data související s místem prostřednictvím aplikací s prostorovými funkcemi, což v konečném důsledku usnadňuje koncovým uživatelům správné rozhodování. [2]

## SQL Express verze 2008

Microsoft SQL Server 2008 Express je “volná”, tedy zdarma nabízená varianta SQL Serveru, ideální pro výuku, vývoj a běh desktopových, webových a malých serverových aplikací.

Stejně jako plná verze SQL Serveru nabízí:

- podporu uložených procedur, triggerů, funkcí, a VIEW tabulek
- ukládání všech typů dat s úplnou podporou pro relační data, XML, FILESTREAM a prostorová data
- vylepšený výkon, použití, vizualizace a navíc integraci s Microsoft 2007 Office System pomocí služeb SQL Server Reporting Services
- integraci s produkty Visual Studio a Visual Web Developer

## Vybrané vlastnosti SQL Express:

**Mocný systém** - vývoj vysoce výkonných aplikací pomocí zabudovaného databázového stroje SQL Serveru, který poskytuje bohaté funkce v kompaktním celku

- Tvorba robustních aplikací
  - vytvoření desktopových a serverových aplikací pomocí pokročilých funkcí SQL Serveru
  - centralizace obchodní logiky a zvýšení výkonu aplikací pomocí uložených procedur, CLR integrace, triggerů a pohledů
  - tvorba aplikací, které dynamicky reagují na změny dat pomocí dotazů (QUERY)
- Správa všech typů dat



- využití všech datových typů SQL Serveru, včetně podpory strukturovaných a nestrukturovaných dat
- ukládání všech typů podnikových dat s nativní podporu pro relační data, XML a prostorová data
- ukládání a validace XML dat a využití XQuery k načtení dat
- ukládání velkých multimediálních souborů v souborových systémech, jako kdyby byly uloženy v databázi pomocí FILESTREAM, soubory se nepočítají do 4GB limitu velikosti databáze
- Mapování klientských dat
  - přidání zeměpisných informací pro klientských aplikací

**Produktivní systém** – zjednodušení vývoje použitím Transact-SQL a nových technologií včetně ADO.NET Entity Frameworku a Language Integrated Query (LINQ).

- Využití stávajících dovedností
  - řešení komplexních obchodních a aplikačních potřeb s plnou podporou pro Transact-SQL
  - tvorba inteligentní rutin pro přístup k datům pomocí uložených procedur a uživatelsky definovaných funkcí
  - odezva na uživatelskou aktivitu a změnu dat pomocí TRIGGERů
  - zvýšení výkonu s přetrvávající dotazovací logikou pomocí VIEW
- Urychlení vývoje
  - modelování dat pomocí nového ADO.NET Entity Frameworku pro skrytí detailů o schématu databáze a přístup k datům pomocí subjektů, které úzce napodobují obchodní logiku
  - podpora LINQ, včetně LINQ to SQL a LINQ to Entities, umožňuje získání dat od subjektů nativně z jakéhokoliv jazyka .NET
- Zlepšení produktivity vývojářů
  - zvýšení produktivity vývojářů a zjednodušení nasazení úsilí pomocí integrace SQL Server Express se všemi edicemi Visual Studio 2008 a Visual Web Developer
  - návrh reálných tabulek a dotazů
  - plná integrace se Server Explorerem



- snadné nasazení ClickOnce
- design s lehkostí

**Pružný systém** - vývoj aplikací, které dokážou řídit všechny typy dat odkudkoliv a uchovávají data snadno synchronizována

- Přístup k datům odkudkoliv
  - tvorba online aplikací, které mohou fungovat i při odpojení od sítě
  - vytvoření desktopové aplikace, která umožňuje automatickou synchronizaci s centrální databází
- Zjednodušení vzdálené správy
  - správa podnikových datových infrastruktur na základě politiky přístupových práv a použití této politiky na instalace SQL Server Expressu
  - snížení času potřebného na denní údržbu definováním společných politik pro databázové operace, jako jsou optimalizace dotazů, jmenné konvence, operací zálohování a obnovy provozu a managementu indexování
  - použití centrální politiky na více serverů pro konzistentní řízení v rámci celého podniku
- Synchronizace firemních dat
  - udržení dat synchronizovaných s pomocí podpory v SQL Server umožňující transakční a slučovací replikace
  - využití pokročilých funkcí řešení konfliktu podporovaných SQL Server Expressem pro vybudování silného podnikového řešení
  - synchronizování dat automaticky s podporou SQL Server Expressu pro sledování změn
- Přístup k datům pomocí PHP
  - PHP vývojáři mohou přistupovat k databázím v SQL Server pomocí ovladače SQL Server pro PHP
  - použití jakéhokoliv vývojového prostředí, možnost zápisu PHP kódu ve Visual Studiu



Správa SQL Server Express databází pomocí SQL Server Management Studio:

- Připojení k lokální databázi SQL Server Express
- Správa objektů s plnou integrací Object Exploreru
- Zápis, oprava a testování dotazů pomocí „vizuálních plánů dotazů“, které nabízí vodítka pro správné odladění vašich skriptů
- Možnosti pro údržbu a management přístupů

#### **Tvorba lokalizovaných aplikací**

- Poskytování relevantních informací
- Vizualizace dat pomocí SQL Server Reporting Services, který je součástí SQL Server Express se službou Advanced Services
- Vytvoření čitelných zpráv, které pomáhají uživatelům odpovědět na složité otázky a pochopit data
- Povolit uživatelům sdílet zprávy interně i externě pomocí aplikace Microsoft Office Word a Microsoft Office Excel

**[3]**



## 4. Jazyk PHP a jeho možnosti použití

Dynamické stránky (tj. stránky nejprve vygenerované serverem a poté odeslané klientovi) jsou v současné době nezbytnou součástí každé složitější internetové prezentace či jen trochu složitější webové stránky.

K hlavním skriptovacím jazykům, které se používají k tvorbě těchto stránek, patří ASP (Active Server Pages) a PHP. Právě jazyku PHP se budu věnován na následujících řádcích.

PHP je programovací jazyk umožňující procedurální i objektově orientované programování. Znalost objektově orientovaného programování (OOP) tedy může být při práci v PHP výhodou, není však nutnou podmínkou. PHP také patří mezi jazyky, kde například není nutné předem definovat typ proměnných (PHP si je dokáže sám typovat dle typu vložených dat), navíc jakákoli proměnná může kdykoli změnit svůj typ. Jednoduše řečeno, co se týče psaní kódu, z PHP při psaní skriptů "sálá" určitá volnost a neomezenost. Na druhé straně záleží plně na programátorovi, jaký si bude v kódu udržovat pořádek.

PHP je velmi univerzální jazyk, ve kterém lze poměrně lehce naprogramovat třeba téměř jakoukoliv aplikaci, např. evidenční databázi programového vybavení, nebo diskusní fórum (jako příklad bych uvedl poměrně oblíbený produkt phpBB). Data lze ukládat buď do obyčejných textových souborů nebo do většiny databází (PHP si dobře rozumí s většinou běžně používaných databází, jmenujme alespoň populární „free databázi“ MySQL či profesionálnější MSSQL). Výbornou vlastností je zpracovávání dat z formulářů, snadno lze vytvořit různé dotazníky, včetně statistik či aplikaci pro správu uživatelů s možností editace atd. O síle PHP svědčí jeho používání na serverech Email.cz, Centrum.cz či BillBoard.cz

Jazyk PHP je v základu technologie umožňující snadné programování na straně serveru (server-side programming). Toho lze využít např. k tvorbě různých interaktivních webových stránek. Stručně lze říci, že skript napsaný v PHP je proveden na serveru podle zadaných kritérií a výsledek je odeslán volajícímu počítači stejným způsobem, jakým se odesílají běžné statické (HTML či XHTML) stránky. Tuto stránku (již bez příkazů PHP) pak dokáže zobrazit jakýkoliv webový prohlížeč. Toto je další dobrá vlastnost PHP - na rozdíl od client-side jazyků,



jako je třeba JavaScript, se k vašemu pracně vytvořenému kódu nikdo nedostane a nemůže ho tak lehce okopírovat.

Server může PHP skripty hledat ve všech odesílaných souborech, ale zpravidla je nakonfigurován tak, aby je hledal v souborech s příponami .php, .php3 nebo .phtml. Příkazy PHP jsou vkládány přímo do HTML kódu a jsou od něj odděleny tagy `<? a ?>` (nebo `<?php a ?>`).

Jakmile je však stránka načtena klientem, pomocí PHP ji již není možné dále měnit, neboť do klienta přichází již čisté HTML.

PHP se vkládá do HTML, takže k tvorbě PHP skriptů obvykle postačuje jakýkoliv běžný HTML editor. Já osobně při tvorbě obyčejných HTML stránek nebo PHP skriptů zcela vystačím s programem PSPAD. Nejdůležitější je ale mít webový server s podporou PHP. Nejvýkonnější je PHP jako modul serveru Apache pod operačním systémem Linux, ale lze ho používat i pod Windows a to buď s webovým serverem IIS nebo opět s Apachem. Podpora PHP patří k jedné ze základních vlastností téměř jakéhokoliv webhostingu, a to buď placeného nebo zdarma. Možnost použití databází je již věc jiná, na většině hostingů s podporou SQL bývá v 99.9% případech podpora MySQL.

Pro ilustraci zde uvádím jednoduchý skript, který vypíše aktuální čas:

```
<html>
<head>
<title>Ukázkový skript PHP</title>
</head>
Aktuální čas: <?php echo Date ("H:i:s"); ?>
</body>
</html>
```

### Jak skript přesně pracuje?

Důležité je si všimnout, že jde v obyčejnou HTML stránku, která obsahuje jeden PHP příkaz, a to ***echo Date ("H:i:s")***, který je od okolního HTML kódu oddělen značkami `<?php a ?>`. Server nejprve vezme soubor HTML, dle přípony php pozná že se jedná o PHP skript, a proto ho nejprve prožene interpretem PHP a vykoná všechny příkazy - ty hledá právě mezi oddělovacími značkami `<?php a ?>`. Narazí na příkaz ***echo***, který slouží k zápisu





do výsledného souboru. Funkce **Date** vrací datum a čas, v závorce jsou její parametry, které v tomto případě určují formát, ve kterém má být čas zobrazen (*hodiny, minuty a sekundy oddělené dvojtečkami*). Následující středník slouží k oddělení více příkazů od sebe. Interpret PHP tedy namísto příkazu PHP zapíše do stránky aktuální čas a server poté stránku odešle do webového prohlížeče klienta. Pokud si zobrazíte zdroj výsledku skriptu, opravdu uvidíte pouze čisté HTML, žádné PHP. **[4]**

**Výsledný soubor tedy vypadá takto:**

```
<html>
<head>
<title>Ukázkový skript PHP</title>
</head>
Aktuální čas: 11:36:31</body>
</html>
```

**Nyní něco o historii PHP**

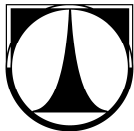
Počátky vývoje **PHP** sahají do roku 1994, kdy Rasmus Lerdorf vytvořil jednoduchý systém evidence přístupů ke svému webu, nejdříve v jazyce Perl, poté v C. Tento systém se rozšířil mezi další uživatele, kteří přicházeli s požadavky na vylepšení. Vznikl tak systém Personal Home Page Tools, později Personal Home Page Construction Kit.

**Vývojové verze:**

**PHP/FI** – z roku 1995 se dá považovat za první „oficiální“ verzi PHP. Byla velmi podobná jazyku Perl, měla jeho proměnné, ale byla mnohem omezenější a jednodušší. Rasmus Lerdorf vytvořil i nástroj umožňující začleňování SQL dotazů a tím zpřístupnění databází na serveru – Forms Interpreter (**FI**).

**PHP/FI 2.0** - roku 1997 bylo oficiálně uvolněno PHP/FI 2.0, jednalo se o verzi napsanou v jazyce C. Tato verze se stala kultovní záležitostí pro tisíce uživatelů na celém světě. Uživatelé mohli přidávat své části kódu. Většinu svého života strávilo v beta verzi, a chvíli po oficiálním uvolnění vznikla alfa verze **PHP 3.0**.

**PHP 3** – vytvořeno v roce 1997 dvojicí Andi Gutmans a Zeev Suraski. Jedná se o kompletně přepracované PHP/FI 2.0, které bylo poddimenzované pro jejich projekty. Původní zkratka dostává nový význam – PHP: Hypertext Preprocessor. Hlavní zbraní této verze bylo obrovské



množství rozšíření a podpora mnoha databází. Dalším klíčovým prvkem byla podpora objektově orientované syntaxe a mnohem silnější syntaxe jazyka.

**PHP 4** - PHP 4.0, oficiálně uvolněné v roce 2000 je výkonnější verzí než PHP 3, přidává například podporu pro mnoho WWW serverů, HTTP sessions, buffering výstupu, bezpečnější způsoby zpracování vstupů uživatele a nové jazykové konstrukty. Jeho engine nese název „Zend Engine“. Dnešní komunita uživatelů PHP zahrnuje stovky tisíc vývojářů a PHP běží na několika milionech serverů.

**PHP 5** - V červnu 2003 byla oficiálně uvolněna betaverze PHP 5, následovaná v červenci 2004 ostrou verzí. Je řízena novým jádrem „Zend Engine 2.0“. Největší změna je v objektovém modelu, PHP se přibližuje ostatním jazykům podporujícím OOP, dále zlepšuje práci s XML a MySQL. Rozšíření pro práci s MySQL bylo kompletně přepsáno. Mezi dalšími novými funkcemi patří zmínit implementace SOAP, použití iterátorů pro zjednodušení pohledů na data a podpora embedded databáze SQLite. Snazší je také obsluha chyb pomocí ošetřování výjimek.

[5]

Poslední verze je verze 5.3.2 uvolněná dne 04.03.2010

PHP lze použít ve všech rozšířených operačních systémech a je podporováno většinou webových serverů.

### **Podporované operační systémy**

Linux, UNIX, Solaris a OpenBSD, Microsoft Windows, Mac OS X, RISC OS, ...

### **Podporované webové servery**

Apache, Microsoft Internet Information Server (IIS), Personal Web Server, servery Netscape a Planet, Oreilly Website Pro, Caudium, Xitami, ...

### **Podpora databází v PHP**

V současné době jsou podporovány všechny „velké“ databáze a samozřejmě mnoho dalších: *Interbase, PostgreSQL, MySQL, MSSQL, Oracle, Informix, Sybase*, atd...

**Co budeme potřebovat k psaní PHP skriptů**

K programování v PHP budeme potřebovat vlastní aplikaci PHP, webový server (Apache nebo jiný webový server – třeba IIS standardně zabudovaný ve Windows), editor (buď přímo HTML nebo jakýkoliv textový), PHP manuál (alespoň pro začátek nutný!) a pro práci s databázemi i MySQL (nebo jakoukoliv jinou podporovanou databázi). Vše prvně jmenované je volně stažitelné z internetu.

**PHP** - můžeme stáhnout na oficiálním serveru projektu: [www.php.net](http://www.php.net).

**Apache** - webový server, na kterém PHP skripty běží. Ke stažení je na [www.apache.org](http://www.apache.org).

**Editor** - pro psaní PHP skriptů si vystačíme s obyčejným HTML editorem. Většina z nich s PHP počítá a skriptovací část PHP je od HTML kódu vizuálně odlišena. Já osobně používám program PSPAD (<http://www.pspad.com/>). Pokud se nechceme spokojit s „málem“, které HTML editory nabízejí, existují editory specializované na PHP. Ty pak nabízí další možnosti.

**PHP manuál** - do PHP manuálu se budeme koukat nejdříve, když něco nebudeme vědět ;-)  
<http://cz.php.net/manual/en/index.php>

**MySQL** - pokud budeme chtít data ukládat do databáze, potřebujeme MySQL. Stáhnout ho můžeme na <http://dev.mysql.com/downloads>.



## 5. Tisková knihovna FPDF

FPDF je PHP knihovna, která umožňuje tvorbu PDF souborů přímo z PHP aplikace. Písmeno F v názvu FPDF znamená Free (Zdarma), je možno ji používat pro jakýkoliv účel a měnit ji, tak aby vyhovovala požadavkům aplikace.

Hlavními rysy FPDF jsou:

- Možnost volby jednotek, formátu papíru a okrajů
- Použití záhlaví a zápatí stránky
- Automaticky vkládaná nová stránka
- Automatické řádky a zarovnání
- Podpora obrázkových souborů (JPEG, PNG a GIF)
- Použití barev (pro písmo, výplně, atd.)
- Možnost vkládání hypertextových odkazů
- Podpora TrueType, Type1 a podpora kódování (včetně českých znaků)
- Účinná komprese stránek

Knihovna FPDF pracuje s PHP4 a PHP5 (FPDF do verze 1.4 pracuje také s PHP3) a nevyžaduje žádné rozšíření (s výjimkou knihovny "zlib" pro aktivaci komprese a knihovny "GD" pro podporu GIF).

Knihovna dokáže vytvářet dokumenty v mnoha jazycích včetně mimo-evropských. Jsou to jazyky středoevropské, azbuka, řecké, baltské a thajské (za předpokladu, že máme vlastní, Type1 nebo TrueType písmo s požadovanou znakovou sadu). Dále je vložena podpora pro čínská, japonská a korejská písma.

Rychlost generování PDF dokumentu je nižší než u knihovny PDFLib, protože FPDF generuje PDF dokument přímo z PHP a ne z předkompilované knihovny. Nicméně, výkon je velmi rozumný a ve většině případů vyhovuje (pokud nejsou dokumenty obzvlášť složité). **[6]**

Použití knihovny je poměrně jednoduché, jak ilustruji na následujícím příkladu a části mého tiskového skriptu. Jediné čeho je třeba se vyvarovat je výpis jakéhokoliv textu na obrazovku po volání funkce *Output()*. V tomto případě nedojde k vygenerování PDF a program zahlásí chybu.



## Základní příkazy FPDF

Add Page() – přidání nové stránky

AddFont – přidá nový font pro použití v textu

SetFont(font,řez,velikost) – nastavení písma dle nastavení

Cell(délka,výška,text,okraje,pozice po buňce,zarovnání,výplň,odkaz) – buňka (řádek) dle nastavení

Image(soubor, pozice X levého horního okraje, pozice Y levého horního okraje,šířka,výška,formát,odkaz) – obrázek zadaných vlastností

Multicell(šířka, výška, text,okraje,zarovnání,výplň) – víceřádkový text

Output() – uloží nebo odešle soubor

Text(x,y,text) – vypíše zadaný text na pozici x,y

Ln(výška) – vloží prázdný řádek o zadané výšce

## Ukázka použití FPDF

```
<?php
require( 'fpdf.php' );
$pdf=new FPDF();
$pdf->AddPage();
$pdf->SetFont( 'Times', 'B', 16 );
$pdf->Cell(40,10, 'Tisková knihovna FPDF' );
$pdf->Output();
?>
```

## Popis jednotlivých příkazů

```
<?php ?>
```

- uvozující a ukončovací PHP tag

```
require( 'fpdf.php' );
```

- nastavení knihovny FPDF

```
$pdf=new FPDF();
```

- nový FPDF dokument

```
$pdf->AddPage();
```

- začátek na nové stránce

```
$pdf->SetFont( 'Times', 'B', 16 );
```

- nastavení písma

```
$pdf->Cell(40,10, 'Tisková knihovna FPDF' );
```

- buňka (řádek) o velikosti 40 a výšce 10

```
$pdf->Output();
```

- tvorba PDF souboru

## 6. Instalace aplikace

Instalace spočívá v rozbalení zdrojových souborů do vybraného adresáře, nastavení webového serveru, tvorbě tabulek a přístupových práv v SQL, nahrání základní skupiny dat do SQL a editaci souboru *init.php*.

Uvádím zde instalaci na počítač vybavený operačním systémem Windows, webovým serverem IIS a databázovým serverem MSSQL a aplikačním serverem PHP 5.0

### Rozbalení souborů

Všechny soubory je nutno nakopírovat do vybrané složky na aplikačním serveru. Obvykle se používá adresář *evidence\_pc* vytvořený ve složce *c:\INETPUB\WWWROOT*, ale je možno použít jakýkoliv adresář.

### Nastavení webového serveru



Obrázek 2: Nastavení IIS

V administraci IIS nastavíme na adresáři s rozbalenými soubory „default“ dokument na soubor *index.php*

### Tvorba tabulek v MSSQL

V administraci tohoto serveru vytvoříme databázi, která bude obsahovat naše tabulky. Doporučuji použít jméno *evidence\_pc*, ale je možno použít jakékoliv které se nevyskytuje v SQL.

Na této databázi pustíme skripty, které jsou na instalačním CDROM v adresáři SQL a na konci tohoto textu (*skripty.sql*). Tyto skripty vytvoří tabulky, uživatele databáze, zapíše první data a vytvoří admin účet pro aplikaci. Pro první přihlášení je vytvořen účet s přihlašovacím



jménem ***prvni\_uzivatel*** a heslem ***Uziv1234***. Pomocí tohoto účtu je vhodné vytvořit ostatní uživatele a následně tento účet z důvodu ochrany přístupu smazat.

**Editace init.php:**

Nakonec je v souboru `init.php` v programové složce nutno upravit řádky s připojením k databázi dle skutečného stavu.

## 7. Vlastní návrh aplikace

### 7.1 Přihlašovací obrazovka



Obrázek 3: Přihlašování do aplikace

Na této obrazovce se musí každý uživatel s právem přístupu přihlásit, každý pokus se zapisuje do LOGu s datumem/časem a přihlašovacím jménem. Pokud bylo zadáno špatné heslo tak i s tímto heslem. Do tohoto LOG souboru má přístup uživatel s právem administrátor ze stránky Administrace uživatelů.

### 7.2 Evidence přihlášení uživatelů

**Evidence přihlášení uživatelů:**  
Omezení výpisu roků: 2008 - 2009 - všechny roky

Kód záznamu	Login	Datum	Čas	Heslo
32	admin	23.03.2009	10:31:36	admin
33	admin	23.03.2009	10:31:19	OK
34	honza	23.03.2009	10:32:38	OK
35	admin	23.03.2009	10:32:52	OK
36	admin	23.03.2009	11:47:12	OK
37	admin	23.03.2009	13:44:57	OK
38	admin	23.03.2009	13:51:28	OK
39	admin	23.03.2009	15:01:10	OK
40	admin	25.03.2009	10:54:27	OK
41	admin	26.03.2009	12:29:59	OK
42	admin	03.04.2009	09:11:01	OK
43	admin	26.05.2009	09:03:25	OK
44	admin	26.05.2009	09:23:37	OK
45	admin	01.06.2009	08:28:41	OK
46	admin	01.06.2009	08:31:30	OK
47	admin	01.06.2009	08:32:23	OK
48	honza	01.06.2009	08:32:35	OK
49	admin	01.06.2009	08:32:43	OK
50	projekt	01.06.2009	08:33:29	OK
51	admin	01.06.2009	12:53:22	OK

Obrázek 4: Evidence přihlášení





Soubor evidující přihlášení uživatelů. První řádek ukazuje příklad chybného pokusu o přístup i s uvedením zadaného hesla. Každý přístup se zapisuje ve tvaru pořadové číslo přístupu, použité přihlašovací jméno, datum a čas, a buď kód OK v případě oprávněného přístupu, nebo v případě pokusu o neoprávněný přístup je zde uvedeno použité heslo. Takový řádek je červeně zvýrazněn.

### 7.3 Úvodní obrazovka

The screenshot shows a web application interface for managing software. At the top, there are navigation buttons: Správa PC, Správa programů, Propojení PC x SOFT, Tisková sestava, Správa uživatelů, and Znovu se přihlásit. The main title is 'Evidence programových aplikací'. Below it, there are buttons for 'Přidat program', 'Měnit záznamy:', 'Editovat program', 'Přidat upgrade', 'Reload', and 'Smazat'. The main section is titled 'Seznam aplikací - 122 záznamů'. It displays a detailed view of a specific application with the following data:

Pořadové číslo:	1	Inventární číslo:	
Název programu:	Windows Commander		
Verze počítačů:	5.0	Verze aktuální:	7.04a
Autor programu:	Christian Ghisler		
Dodavatel:	AutoCont a.s. Karlovy Vary		
Smluvní závazky:			
Typ licence:	freeware	Typ programu:	osobní stanice
Počet licencí:	13	Správce:	
Poznámka:	produkt přejmenován na Total Commander		
Popis programu:	Souborový manažer		

Below the application details, there are buttons for 'Předchozí záznam (č.1)', 'Jdi na záznam číslo', and 'Další záznam (č.2)'. At the bottom, there is a section for 'Update a upgrade:' with a table showing the following data:

Faktura:	Verze:	Licence:	Poznámka:	Datum:
913/01	6.52	8	Free upgrade na Total Commander	

Below this, there is a section for 'Přiřazená PC:' with a table showing the following data:

ID Počítače	Jméno PC	IP adresa
2	jan.novak	192.168.34.56
3	h	xcbv
4	h	hb
5	655	7

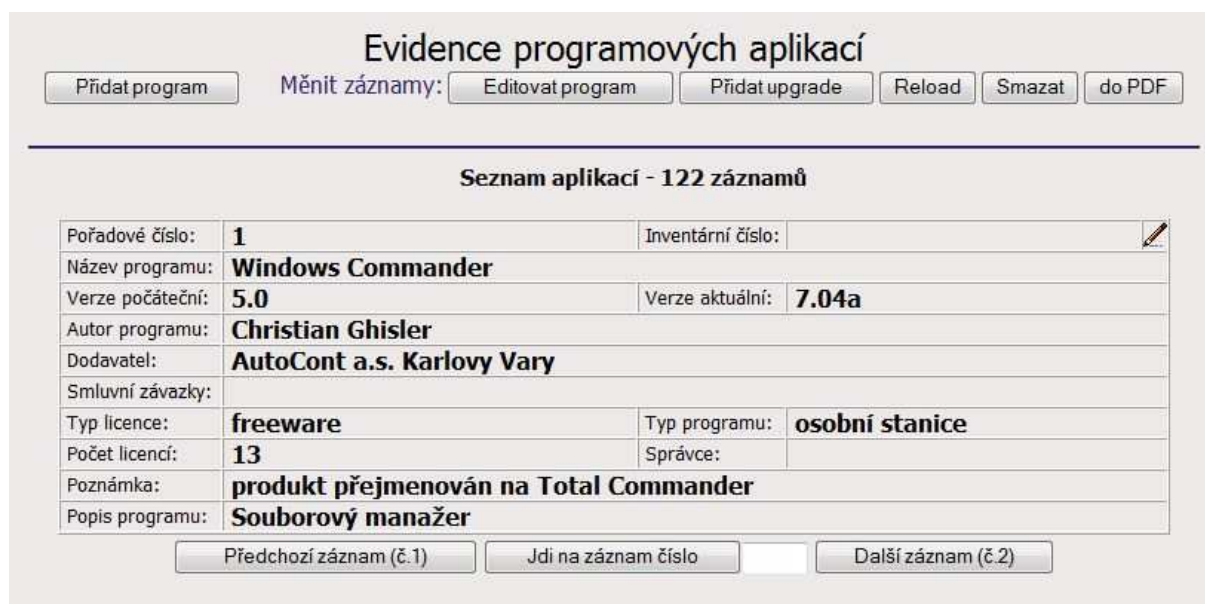
Obrázek 5: Úvodní obrazovka

Nahoře je zobrazena lišta s tlačítky pro ovládání aplikace (pro uživatele s omezeným přístupem jsou určitá tlačítka neaktivní).

Na této základní obrazovce uživatel stiskem příslušného tlačítka přejde na vybranou stránku.

Správa PC – evidence počítačů, Správa programů – evidence aplikací, Propojení PC x SOFT – zobrazení propojení mezi PC a programy, Tisková sestava – export do PDF, Správa uživatelů – nastavení přístupu, Znovu se přihlásit – pro přihlášení do aplikace pod jiným uživatelem.

## 7.4 Evidence programových aplikací



Evidence programových aplikací	
Přidat program	Měnit záznamy: Editovat program Přidat upgrade Reload Smazat do PDF
Seznam aplikací - 122 záznamů	
Pořadové číslo:	1
Inventární číslo:	
Název programu:	Windows Commander
Verze počáteční:	5.0
Verze aktuální:	7.04a
Autor programu:	Christian Ghisler
Dodavatel:	AutoCont a.s. Karlovy Vary
Smluvní závazky:	
Typ licence:	freeware
Typ programu:	osobní stanice
Počet licencí:	13
Správce:	
Poznámka:	produkt přejmenován na Total Commander
Popis programu:	Souborový manažer
Předchozí záznam (č.1) Jdi na záznam číslo Další záznam (č.2)	

Obrázek 6: Okno práce se softwarem

Stránka s zobrazující programovou aplikaci s možností práce se záznamem, listováním, atd.

Zde lze stiskem tlačítka **Přidat program** zadat do databáze další programovou aplikaci.

Tlačítkem **Editovat program** lze provést změnu údajů pro zobrazený program.

Stiskem **Přidat upgrade** lze k zobrazenému programu přidat aktualizaci/update, totéž lze provést kliknutím na zelené plus v části **Upgrade**.

Tlačítko **Reload** provede znovunačtení aktuálního záznamu z databáze, např. po provedené editaci.

Tlačítkem **Smazat** se vymaže vybraná aplikace včetně všech aktualizací a propojení na PC.

Tlačítko **do PDF** převede zobrazený list softwaru do formátu PDF. Listování umožňují tlačítka **Předchozí / Další záznam**. Pole **Jdi na záznam číslo** provede skok na příslušný program.



## 7.5 Evidenční list programové aplikace

Evidenční list programové aplikace

Pořadové číslo: 126

Inventurní číslo:

Název programu:

Verze počáteční:

Verze aktuální:

Autor programu:

Dodavatel:

Smluvní závazky:

Typ licence: uživatelská

Typ programu: ISVS

Počet licencí:

Poznámka:

Správce:

Popis programu:

**Před uložením přkontrolujte správnost údajů**

Uložit

Obrázek 7: Dialogové okno přidání programu

Zde lze zadat do databáze další programovou aplikaci. Je možno volit typ licence (uživatelská, síťová, serverová, multilicence, atd.) a typ programu (externí, provozní, serverový, atd.)

Editace programu používá stejný dialog, pouze s před-vyplněnými údaji dle aktuálního programu.

## 7.6 Evidenční list update/upgrade programové aplikace

Evidenční list update/upgrade programové aplikace

Program: Windows Commander

Pořadové číslo: 117

Faktura:

Verze:

Počet licencí:

Poznámka:

DATUM:

**Před uložením přkontrolujte správnost údajů**

Uložit

Obrázek 8: Dialogové okno přidání upgradu

Zde je možno přidat k aktuálnímu programu upgrade + zadat určitý popis upgradu. Editace opět ve stejném okně pouze s před-vyplněnými údaji.



## 7.7 Tisk stránky do PDF

### Tisková sestava - Software

Program:

Pořadové číslo	1	Inventurní číslo	
Název programu	Windows Commander		
Verze počáteční	5.0	Verze aktuální	7.04a
Autor programu	Christian Ghisler		
Dodavatel	AutoCont a.s. Karlovy Vary		
Smluvní závazky			
Typ licence	freeware	Typ programu	osobní stanice
Počet licencí	13	Správce	
Poznámka	produkt přejmenován na Total Commander		
Popis programu	Souborový manažer		

Update a upgrade:

Faktura	Verze	Licence	Poznámka	Datum
	6.52	8	Free upgrade na Total Commander	
913/01		5		
		0		
		9		

Přiřazená PC:

ID počítače	Jméno PC	IP adresa
2	jan.novak	192.168.34.56
3	h	xcbv
4	h	hb
5	655	7

Obrázek 9: Tisk aktuální stránky do PDF

Zde je zobrazen podoba vytištěné aktuální stránky do PDF.

## 7.8 Update a upgrade

Update a upgrade:					
Faktura:	Verze:	Licence:	Poznámka:	Datum:	
	6.52	8	Free upgrade na Total Commander		
913/01		5			

Přiřazená PC:			
ID Počítače	Jméno PC	IP adresa	
2	jan.novak	192.168.34.56	
3	h	xcbv	
4	h	hb	
5	655	7	

Obrázek 10: Upgrade programu

Seznam upgradu a nových verzí příslušného programu, opět s možností editace.

V polích je zobrazen veškerý zaznamenaný upgrade programu, včetně čísla faktury, aktuální verze, počtu licencí. Každý řádek je možno editovat (nebo smazat) kliknutím na příslušnou ikonu.

Níže je uveden seznam počítačů na kterých je tento program nainstalován.



Každý řádek znamená 1 počítač, kde je program nainstalován, včetně ID počítače, jména PC, IP adresy. Záznamy lze přidávat, popř. mazat.

## 7.9 Evidence uživatelů

Vyberte uživatele, kterého chcete editovat:

Seznam uživatelů

Jméno uživatele	Právo na DB
jirka	editor
honza	čtenář
admin	administrátor

Úprava hesla

Jméno:

Staré heslo:

Nové heslo:

Vložení uživatele

Jméno:

Heslo:

Právo:

Mazání uživatele

Jméno:

Heslo:

Skutečně chcete záznam vymazat?  
**POZOR - nelze zpět**

Obrázek 11: Evidence uživatelů

Výpis všech uživatelů i s jejich právy, dialogy pro úpravu hesla, vložení uživatele a jeho výmaz. Heslo se při ukládání šifruje pomocí MD5 algoritmu.

Uživatel s právem administrátor zde může provádět změny v uživatelských účtech, změny hesla, zakládat nového uživatele s přidělením práv a mazat určitého uživatele.















Aplikace má dle práv uživatelů různé možnosti editace.

- pro uživatele s právem Čtenář nejsou dostupné žádné editační prvky (editace, přidání, výmaz softwaru, PC, propojení, atd.), nemá právo vstupovat do Správy uživatelů.
- uživatel s právem Editor nesmí mazat záznamy, smí je editovat a přidávat, nesmí do Správy uživatelů.



- uživatel s právem Administrátor má plnou kontrolu nad aplikací, včetně mazání záznamů, editace uživatelů atd. Tento uživatel dále smí na prohlížet LOG soubor s evidencí přístupů (dostupné na stránce Evidence uživatelů)

## 7.10 Evidence PC

Evidence PC					
Pořadové číslo	Inventární číslo	Umístění	Jméno	IP	
1	KUKVdd15	B123	tomas.bucek	192.168.34.42	 
2	dggfg	b452	jan.novak	192.168.34.56	 
3	fgn	hh	h	xcvv	 
4	b	hb	h	hb	 
5	44	4	655	7	 
6	hjjcj	145	TOMAS.	781.45.874.2577	 
7	787	545	5454	5454	 

(c) 2005 - oddělení GIS Karlovarského kraje

Obrázek 12: Evidence PC

Seznam PC s možností přidání, editace, výmazu.

Pro každý PC je zadáno jeho pořadové číslo, inventární číslo, umístění, jméno a IP adresa. Záznamy lze opět přidávat stisknutím zeleného plus, editovat (mazat) příslušný záznam pomocí grafických ikon. Možnost použití těchto editačních prvků je dána příslušnými právy přihlášeného uživatele.

## 7.11 Přidání / editace PC

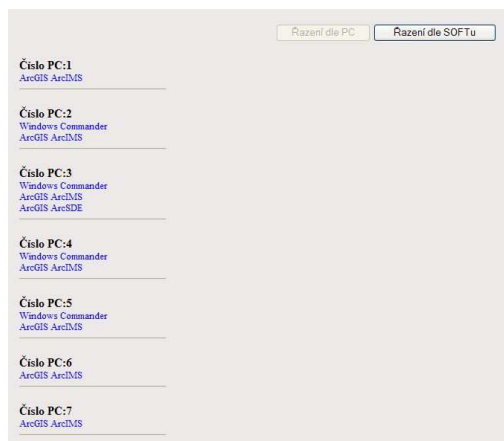
Evidenční list PC	
Pořadové číslo: 8	
Inv. číslo:	<input type="text"/>
Umístění:	<input type="text"/>
Jméno:	<input type="text"/>
IP:	<input type="text"/>
<input type="button" value="Uložit"/>	

Obrázek 13: Přidání PC

Dialogové okno pro přidání PC do evidence. Pro editaci se opět používá stejný dialog, pouze s vyplněnými údaji. Zadává se inventární číslo PC, dále jeho umístění, jméno (popis) počítače a nakonec IP adresa.



## 7.12 Přiřazení PC k software



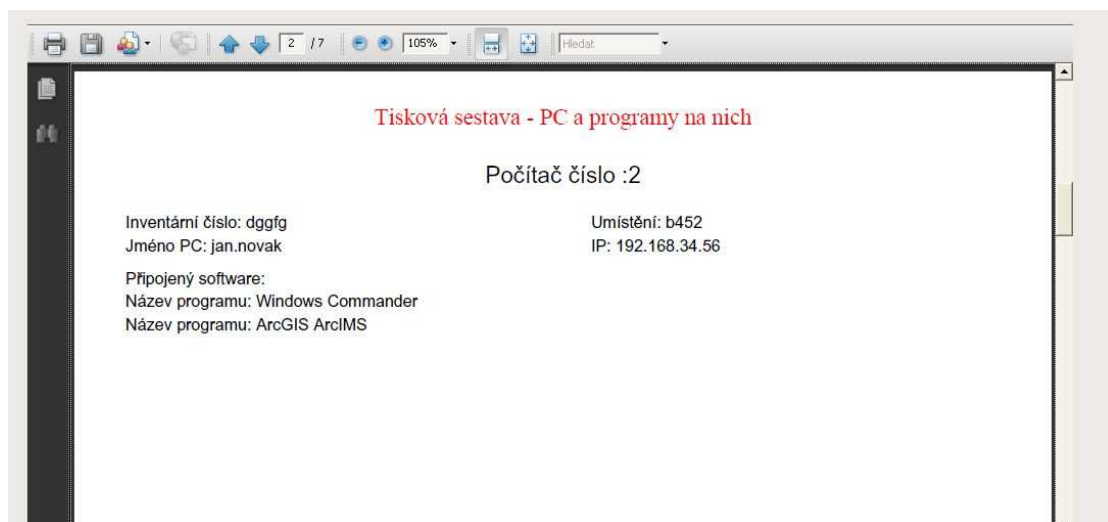
Obrázek 15: Přiřazení PC – řazení dle software

Obrázek 14: Přiřazení PC – řazení dle PC

Dva pohledy na část přiřazení PC k programu, nejprve seznam PC a programů na nich, poté seznam programů a PC na kterých ten který je.

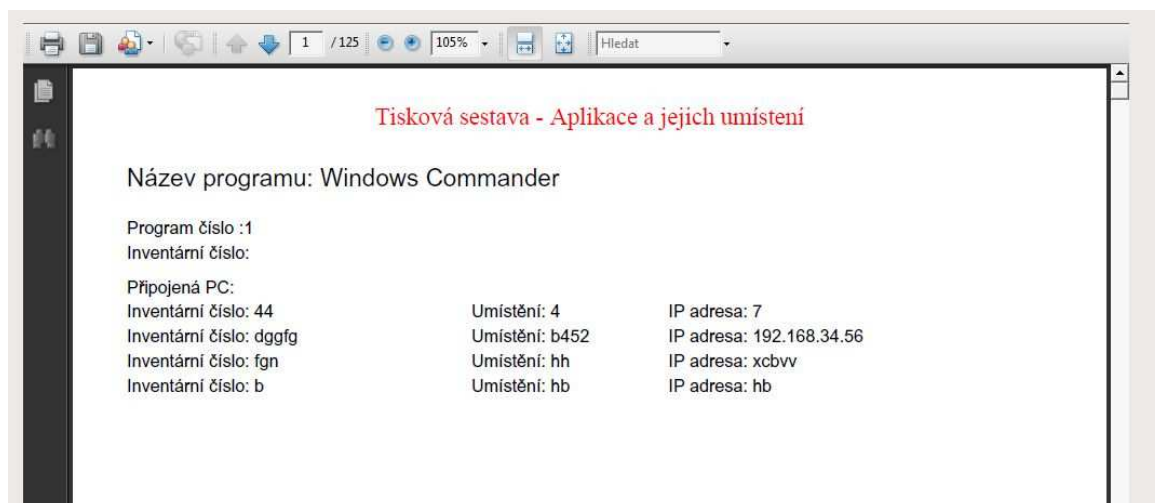
V případě výpisu dle PC je po kliknutí na název softwaru uživatel přenesen na evidenční list příslušného softwaru.

## 7.13 Tisková sestava



Obrázek 16: Tisková sestava v PDF – řazení dle PC

Ukázka tiskové sestavy, řazení dle PC.



**Obrázek 17: Tisková sestava v PDF – řazení dle softwaru**

Ukázka tiskové sestavy, řazení dle software.

Výsledný tvar PDF souboru lze upravovat editací příslušného skriptu (tisk\_soft.php resp. tisk\_pc.php). Ukázku tohoto tiskového skriptu můžete nalézt o pár stránek dál.





## 8. Přílohy

### 8.1 Struktura databáze MSSQL

Name	Owner	Type	Create Date
apl_upd	dbo	User	3.11.2008 9:28:21
aplikace	dbo	User	3.11.2008 9:28:21
dtproperties	dbo	System	3.11.2008 9:28:35
LOG_NAVSTEV	dbo	User	3.11.2008 9:38:16
login	dbo	User	3.11.2008 10:48:59
pc_to_soft	dbo	User	3.12.2008 14:56:37
pocitace	dbo	User	3.11.2008 9:30:15
syscolumns	dbo	System	6.8.2000 1:29:12
syscomments	dbo	System	6.8.2000 1:29:12
sysdepends	dbo	System	6.8.2000 1:29:12
sysfilegroups	dbo	System	6.8.2000 1:29:12
sysfiles	dbo	System	6.8.2000 1:29:12
sysfiles1	dbo	System	6.8.2000 1:29:12
sysforeignkeys	dbo	System	6.8.2000 1:29:12
sysfulltextcatalogs	dbo	System	6.8.2000 1:29:12
sysfulltextnotify	dbo	System	6.8.2000 1:29:12
sysindexes	dbo	System	6.8.2000 1:29:12
sysindexkeys	dbo	System	6.8.2000 1:29:12
sysmembers	dbo	System	6.8.2000 1:29:12
sysobjects	dbo	System	6.8.2000 1:29:12
syspermissions	dbo	System	6.8.2000 1:29:12
sysproperties	dbo	System	6.8.2000 1:29:12
sysprotects	dbo	System	6.8.2000 1:29:12
sysreferences	dbo	System	6.8.2000 1:29:12
systypes	dbo	System	6.8.2000 1:29:12
sysusers	dbo	System	6.8.2000 1:29:12

Obrázek 18: Tabulky v databázi MSSQL

Pohled na databázi v MSSQL, seznam tabulek

**apl\_upd** – tabulka s evidencí upgradů softwaru

**aplikace** – tabulka s evidencí programů

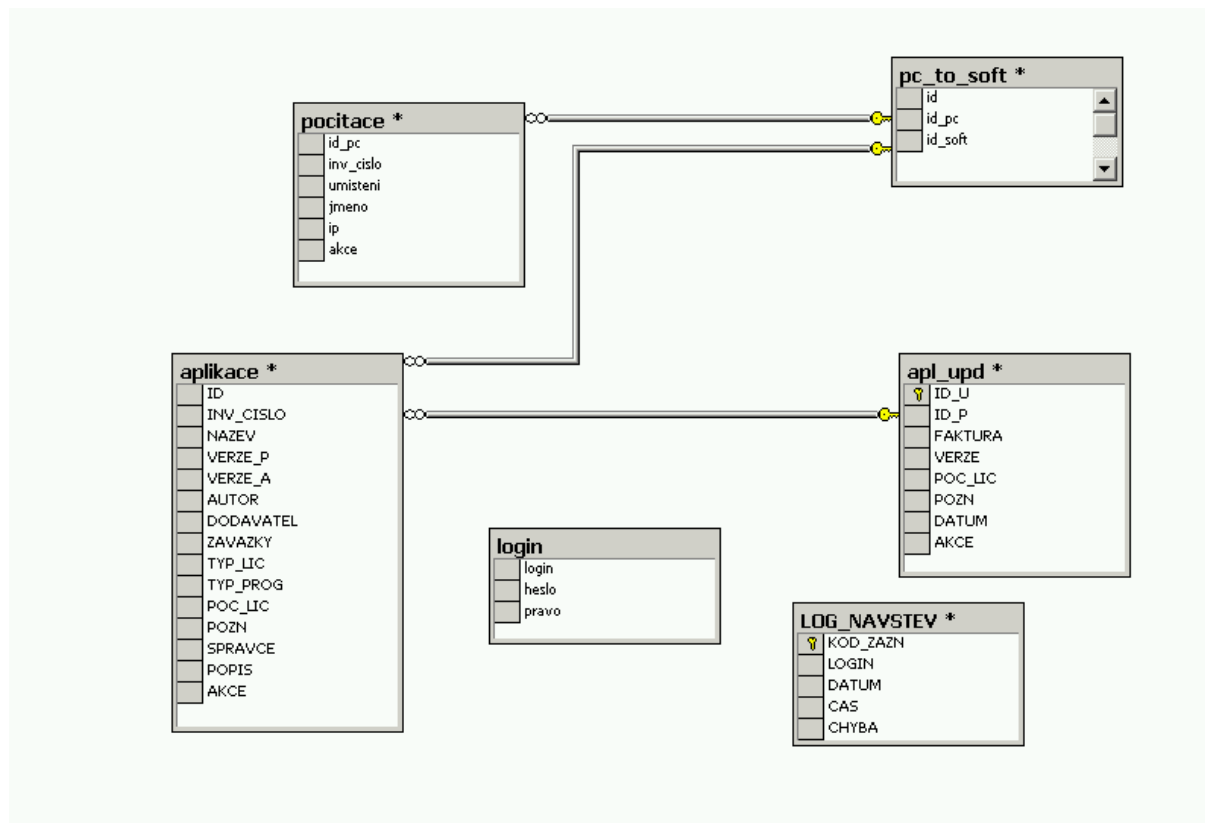
**log\_navstev** – log soubor s přístupy

**login** – evidence uživatelů

**pc\_to\_soft** – propojovací tabulka PC a softwaru

**pocitace** – evidence PC

## 8.2 Schéma databáze



Obrázek 19: Schéma databáze

## 8.3 Výpis skriptů pro vytvoření tabulek:

### Apl\_upd – tabulka upgradu pro software

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[apl_upd]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[apl_upd]
GO
CREATE TABLE [dbo].[apl_upd] (
    [ID_U] [numeric](18, 0) NULL ,
    [ID_P] [numeric](18, 0) NULL ,
    [FAKTURA] [char] (100) COLLATE Czech_CI_AS NULL ,
    [VERZE] [char] (50) COLLATE Czech_CI_AS NULL ,
    [POC_LIC] [numeric](18, 0) NULL ,
    [POZN] [char] (100) COLLATE Czech_CI_AS NULL ,
    [DATUM] [char] (10) COLLATE Czech_CI_AS NULL ,
    [AKCE] [char] (1) COLLATE Czech_CI_AS NULL
) ON [PRIMARY]
GO
```

### Aplikace – tabulka evidence software

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[aplikace]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[aplikace]
GO
CREATE TABLE [dbo].[aplikace] (
    [ID] [numeric](18, 0) NOT NULL ,
    [INV_CISLO] [char] (15) COLLATE Czech_CI_AS NULL ,
    [NAZEV] [char] (200) COLLATE Czech_CI_AS NULL ,
    [VERZE_P] [char] (20) COLLATE Czech_CI_AS NULL ,
    [VERZE_A] [char] (20) COLLATE Czech_CI_AS NULL ,
    [AUTOR] [char] (100) COLLATE Czech_CI_AS NULL ,
    [DODAVATEL] [char] (100) COLLATE Czech_CI_AS NULL ,
```



```
[ZAVAZKY] [char] (100) COLLATE Czech_CI_AS NULL ,
[TYP_LIC] [char] (12) COLLATE Czech_CI_AS NULL ,
[TYP_PROG] [char] (20) COLLATE Czech_CI_AS NULL ,
[POC_LIC] [numeric](18, 0) NULL ,
[POZN] [char] (100) COLLATE Czech_CI_AS NULL ,
[SPRAVCE] [char] (20) COLLATE Czech_CI_AS NULL ,
[POPIS] [char] (300) COLLATE Czech_CI_AS NULL ,
[AKCE] [char] (1) COLLATE Czech_CI_AS NULL
) ON [PRIMARY]
GO
```

### LOG\_NAVSTEV – evidence přístupů do aplikace

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[LOG_NAVSTEV]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[LOG_NAVSTEV]
GO
CREATE TABLE [dbo].[LOG_NAVSTEV] (
[KOD_ZAZN] [numeric](18, 0) NOT NULL ,
[LOGIN] [char] (50) COLLATE Czech_CI_AS NULL ,
[DATUM] [char] (10) COLLATE Czech_CI_AS NULL ,
[CAS] [char] (10) COLLATE Czech_CI_AS NULL ,
[CHYBA] [char] (50) COLLATE Czech_CI_AS NULL
) ON [PRIMARY]
GO
```

### Login – tabulka s uživateli

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[login]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[login]
GO
CREATE TABLE [dbo].[login] (
[login] [char] (50) COLLATE Czech_CI_AS NULL ,
[heslo] [char] (50) COLLATE Czech_CI_AS NULL ,
[pravo] [char] (20) COLLATE Czech_CI_AS NULL
) ON [PRIMARY]
GO
```

### Pc\_to\_soft – tabulka propojení PC s software

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[pc_to_soft]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[pc_to_soft]
GO
CREATE TABLE [dbo].[pc_to_soft] (
[id] [int] IDENTITY (1, 1) NOT NULL ,
[id_pc] [char] (10) COLLATE Czech_CI_AS NULL ,
[id_soft] [char] (10) COLLATE Czech_CI_AS NULL
) ON [PRIMARY]
GO
```

### Pocitace – tabulka evidence PC

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[pocitace]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[pocitace]
GO
CREATE TABLE [dbo].[pocitace] (
[id_pc] [int] NULL ,
[inv_cislo] [char] (50) COLLATE Czech_CI_AS NULL ,
[umisteni] [char] (10) COLLATE Czech_CI_AS NULL ,
[jmeno] [char] (50) COLLATE Czech_CI_AS NULL ,
[ip] [char] (15) COLLATE Czech_CI_AS NULL ,
[akce] [char] (1) COLLATE Czech_CI_AS NULL
) ON [PRIMARY]
GO
```



**Možnost vyzkoušení:**

Aplikace běží na adrese [http://gis.kr-karlovarsky.cz/evidence\\_pc/](http://gis.kr-karlovarsky.cz/evidence_pc/)

Je možno použít login „projekt“ a heslo „projekt“

Přílohou této bakalářské práce je CD s instalačními soubory a skripty pro vytvoření struktury v SQL a nahrání prvotních dat.

## **Seznam obrázků:**

Obrázek 1: Rozložení datové platformy serveru SQL Server 2005

Obrázek 2: Nastavení IIS

Obrázek 3: Přihlašování do aplikace

Obrázek 4: Evidence přihlášení

Obrázek 5: Úvodní obrazovka

Obrázek 6: Okno práce se softwarem

Obrázek 7: Dialogové okno přidání programu

Obrázek 8: Dialogové okno přidání upgradu

Obrázek 9: Tisk aktuální stránky do PDF

Obrázek 10: Upgrade programu

Obrázek 11: Evidence uživatelů

Obrázek 12: Evidence PC

Obrázek 13: Přidání PC

Obrázek 14: Přiřazení PC – řazení dle PC

Obrázek 15: Přiřazení PC – řazení dle software

Obrázek 16: Tisková sestava v PDF – řazení dle PC

Obrázek 17: Tisková sestava v PDF – řazení dle softwaru

Obrázek 18: Tabulky v databázi MSSQL

Obrázek 19: Schéma databáze



## 9. Ukázkový popis skriptů

### 9.1 Skript pro zobrazení aplikace

```
<html>
<head>
<meta http-equiv=Content-Type content="text/html; charset=windows-1250">
<title>Evidence programového vybavení</title>
<meta http-equiv="Refresh" content="15"; URL="" />
<link rel="stylesheet" type="text/css" href="style.css">
<SCRIPT LANGUAGE="JavaScript">
<!--//
function otevritOkno() {
window.open('', 'NoveOkno', 'toolbar=0,location=0,scrollbars=0,resizable=0,status=0,width=600,height=200,top=0,left=0');
}
function otevritOkno2() {

window.open('', 'EditDel', 'toolbar=0,location=0,scrollbars=1,resizable=1,status=0,width=1024,height=768,top=100,left=100');
}
//-->
</SCRIPT>
<script language="JavaScript"><!--
var browser = navigator.appName;
var verze = parseInt(navigator.appVersion);
var W=1024;
var H=768;
window.resizeTo(W, H);
</SCRIPT>
</head>
```

- hlavička stránky s Java skripty používanými pro otevírání oken (nastavení velikosti těchto oken)

```
<?
include "init.php";
echo "<body bgcolor=\"#\".$color.\"\" class=\"text1\">";
echo "<center><font size=\"+2\" face=\"Tahoma\">Evidence programových
aplikací</font></center>";
if (!isset($typ)):$typ = "";endif;
if (!isset($vyb_cislo)):$vyb_cislo = "";endif;
do {
```

- nulování používaných proměnných

```
if (!$vyb_cislo):
$vyb_cislo = "1";
endif;
}
while (false) ;
?>
```

- nastavení prvotního zobrazení na záznam číslo 1

```
<center>
<table bgcolor="#ECE9E8" border=0 cellspacing="0" rules=none>
<?
$pocet2 = ODBC_Exec($spojeni, "SELECT * FROM aplikace ORDER by ID");
$pocet_d = ODBC_Exec($spojeni, "SELECT * FROM aplikace where (AKCE='D') ORDER by ID");
$odecet = ODBC_Num_Rows($pocet2)-ODBC_Num_Rows($pocet_d);
- připojení do databáze, zjištění počtu záznamů
```

```
if ($vyb_cislo > ODBC_Num_Rows($pocet2)):
$vyb_cislo = ODBC_Num_Rows($pocet2);
endif;
```

- ochrana před pokusem o zobrazení vyššího čísla záznamu než je nejvyšší číslo v databázi

```
if ($vyb_cislo < '1'):
$vyb_cislo = "1" ;
endif;
```

- ochrana před pokusem o zobrazení záznamů s číslem nižším než je 1

```
if ($smazat=="A"):
$vysledek_vymaz = ODBC_Exec($spojeni, "DELETE from pc_to_soft where (id=$id_vymaz) ");
if (!$vysledek_vymaz):
```





```
<td><form action="tisk_zaznamy.php?ID=<?echo $vyb_cislo?>" target="EditDel"
onClick="otevritOkno2()" method=post>
<INPUT NAME=login TYPE=HIDDEN VALUE="<? echo $login ?>"><button type="submit" Value="do
PDF">do PDF</button></form></td>
</center>
</table>
</td>
```

- horní tlačítková lišta

```
<center>
<hr width=80% color="#2f246c">
<table width="100%"><td valign=top width="*">
<center><h2><font face="Tahoma" color="#0A0A0A">Seznam aplikací -
<?
echo $odecet." záznamů";
echo "</h2>";
?>
```

- zobrazení počtu záznamů

```
<table border="1" cellspacing="0" cellpadding="0" width="750">
<td width="110"><font size="-1">&nbsp;Pořadové číslo:</font></td>
<td width="290"><b>&nbsp;<?echo ODBC_result($vysledek2, "ID")?></b></td>
<td width="100"><font size="-1">&nbsp;&nbsp;&nbsp;Inventární číslo:</font></td>
<td width="250"><b>&nbsp;<?echo ODBC_result($vysledek2, "INV_CISLO")?></b></td>
<td>
<? if ($login<>0)?>
<a href="uprav_zazn.php?ID=<? echo ODBC_result($vysledek2, "ID")?>" target="EditDel"
onClick="otevritOkno2()" title="Editovat aplikaci"></a></td><tr>
<? else: ?>
</td><tr>
<? endif; ?>
<td><font size="-1">&nbsp;Název programu:</font></td>
<td colspan=4><b>&nbsp;<?echo ODBC_result($vysledek2, "NAZEV")?></b></td><tr>
<td><font size="-1">&nbsp;&nbsp;&nbsp;Verze počáteční:</font></td>
<td><b>&nbsp;<?echo ODBC_result($vysledek2, "VERZE_P")?></b></td>
<td><font size="-1">&nbsp;&nbsp;&nbsp;Verze aktuální:</font></td>
<td colspan=2><b>&nbsp;<?echo ODBC_result($vysledek2, "VERZE_A")?></b></td><tr>
<td><font size="-1">&nbsp;&nbsp;&nbsp;Autor programu:</font></td>
<td colspan=4><b>&nbsp;<?echo ODBC_result($vysledek2, "AUTOR")?></b></td><tr>
<td><font size="-1">&nbsp;&nbsp;&nbsp;Dodavatel:</font></td>
<td colspan=4><b>&nbsp;<?echo ODBC_result($vysledek2, "DODAVATEL")?></b></td><tr>
<td><font size="-1">&nbsp;&nbsp;&nbsp;Smluvní závazky:</font></td>
<td colspan=4><b>&nbsp;<?echo ODBC_result($vysledek2, "ZAVAZKY")?></b></td><tr>
<td><font size="-1">&nbsp;&nbsp;&nbsp;Typ licence:</font></td>
<td><b>&nbsp;<?echo ODBC_result($vysledek2, "TYP_LIC")?></b></td>
<td><font size="-1">&nbsp;&nbsp;&nbsp;Typ programu:</font></td>
<td colspan=2><b>&nbsp;<?echo ODBC_result($vysledek2, "TYP_PROG")?></b></td><tr>
<td><font size="-1">&nbsp;&nbsp;&nbsp;Počet licencí:</font></td>
<td><b>&nbsp;<?echo ODBC_result($vysledek2, "POC_LIC")?></b></td>
<td><font size="-1">&nbsp;&nbsp;&nbsp;Správce:</font></td>
<td colspan=2><b>&nbsp;<?echo ODBC_result($vysledek2, "SPRAVCE")?></b></td><tr>
<td><font size="-1">&nbsp;&nbsp;&nbsp;Poznámka:</font></td>
<td colspan=4><b>&nbsp;<?echo ODBC_result($vysledek2, "POZN")?></b></td><tr>
<td><font size="-1">&nbsp;&nbsp;&nbsp;Popis programu:</font></td>
<td colspan=4><b>&nbsp;<?echo ODBC_result($vysledek2, "POPIS")?></b></td><tr>
</table>
```

- vlastní zobrazení záznamu v tabulce

```
<table><td><form action="index_soft.php" method=post>
<INPUT NAME=login TYPE=HIDDEN VALUE="<? echo $login ?>">
<INPUT NAME=typ TYPE=HIDDEN VALUE=minus>
<INPUT NAME=vyb_cislo TYPE=HIDDEN VALUE="<? echo $vyb_cislo-1 ?>">
<Input type="submit" Value="Předchozí záznam (č.<?
if ($vyb_cislo-1 < '1'):
$ukaz_cislo = "2" ;
else:
$ukaz_cislo = $vyb_cislo;
endif;
echo $ukaz_cislo-1
?>
)"></form></td>
<td><form action="index_soft.php" method=post>
<INPUT NAME=login TYPE=HIDDEN VALUE="<? echo $login ?>">
<input type="submit" Value="Jdi na záznam číslo">
<INPUT NAME=typ TYPE=HIDDEN VALUE=skok>
<INPUT NAME=pov_cislo TYPE=HIDDEN VALUE="<? echo $vyb_cislo ?>">
```



```
<INPUT NAME=vyb_cislo size=3 MAXLENGTH=3></form></td>
<td><form action="index_soft.php" method=post>
<INPUT NAME=login TYPE=HIDDEN VALUE="<? echo $login ?>">
<INPUT NAME=typ TYPE=HIDDEN VALUE=plus>
<INPUT NAME=vyb_cislo TYPE=HIDDEN VALUE="<? echo $vyb_cislo+1 ?>">
<Input type="submit" Value="Další záznam (č.<?
if ($vyb_cislo == ODBC_Num_Rows($pocet2)):
$ukaz_cislo = ODBC_Num_Rows($pocet2);
else:
$ukaz_cislo = $vyb_cislo+1;
endif;
echo $ukaz_cislo
?>)"></form><center>
</td></table>
```

- tlačítka pro listování v záznamech

```
<font size="-1">Update a upgrade:</font>
<table cellpadding="0" cellspacing="0" width="850" border=1>
<td width="200"><font size="2"><b><center>Faktura:</center></b></td>
<td width="200"><font size="2"><b><center>Verze:</center></b></td>
<td width="50"><font size="2"><b><center>Licence:</center></b></td>
<td width="250"><font size="2"><b><center>Poznámka:</center></b></td>
<td width="50"><font size="2"><b><center>Datum:</center></b></td></td>
<? if ($login<>0):?>
<a href="pridej_upgr.php?ID_P=<? echo ODBC_result($vysledek2, "ID")?>" target="EditDel"
onClick="otevritOkno2()" title="Přidat upgrade/update"></a></td><tr>
<? else:?>
</td><tr>
<? endif:?>
<?
while (ODBC_Fetch_Row($vysledek3)):
if (ODBC_result($vysledek3, "AKCE")=='D'):
else:
echo "<td><font size=\"2\">&nbsp;". ODBC_result($vysledek3, "FAKTURA")."</td>
<td><font size=\"2\"><center>&nbsp;". ODBC_result($vysledek3, "VERZE")."</td>
<td><font size=\"2\"><center>&nbsp;". ODBC_result($vysledek3, "POC_LIC")."</td>
<td><font size=\"2\">&nbsp;". ODBC_result($vysledek3, "POZN")."</td>
<td><font size=\"2\">&nbsp;". ODBC_result($vysledek3, "DATUM")."</td>";
if ($login<>0):
echo "<td><a href=\"edit_upd.php?ID_U=\".ODBC_result($vysledek3, "ID_U").\"\" target=\"EditDel\"
onClick=\"otevritOkno2()\" title=\"Editovat řádek\"><img src=\"image\edit.gif\"
border=0></a></td>";
else:
echo "<td><center><img src=\"image\\noedit.gif\" border=0></td>";
endif;
if ($login==2):
echo "<td><a href=\"maz_upd.php?ID_U=\".ODBC_result($vysledek3, "ID_U").\"\" target=\"EditDel\"
onClick=\"otevritOkno2()\" title=\"Smazat řádek\"><img src=\"image\del.gif\"
border=0></a></td><tr>";
else:
echo "<td><center><img src=\"image\\nodel.gif\" border=0></a></td><tr>";
endif;
endif;
endwhile;
echo "</table><br>";
```

- zobrazení upgradu pro příslušný záznam

```
$id_aplikace = ODBC_result($vysledek2, "ID");
echo "Přiřazená PC:";

$prir_pc = ODBC_Exec($spojeni, "SELECT * FROM pc_to_soft where id_soft = $id_aplikace ORDER by id_pc");
echo "<table border=1><td bgcolor=\"#bbbbbb\">&nbsp;ID Počítače&nbsp;</td>
<td bgcolor=\"#bbbbbb\">&nbsp;Jméno PC&nbsp;</td>
<td bgcolor=\"#bbbbbb\">IP adresa</td>";
if ($login<>0):
echo "<td><a href=\"prirad_pc.php?id_aplikace=\".$id_aplikace.\"\" target=\"EditDel\"
onClick=\"otevritOkno2()\" title=\"Přiřadit k PC\"><img src=\"image\pridat.gif\"
border=0></a></td><tr>";
else:
echo "<td></td><tr>";
endif;
while (ODBC_Fetch_Row($prir_pc)):
$n_pc = ODBC_result($prir_pc, "ID_PC");
$nalezn_pc = ODBC_Exec($spojeni, "SELECT * FROM pocitace where id_pc = $n_pc ORDER by id_pc");
echo "<td>".ODBC_result($nalezn_pc, "ID_PC")."</td>
```





```
<td>".ODBC_result($nalez_pc, "jmeno")."</td>
<td>".ODBC_result($nalez_pc, "IP")."</td>";

if ($login==admin):
echo "<td><a href=\"index_soft.php?id_vymaz=\".ODBC_result($prir_pc,
\"ID\".\"&id_aplikace=\". $id_aplikace.\"&login=\".$login.\"&smazat=A\" title=\"Smazat řádek\"><img
src=\"image\del.gif\" border=0></a></td><tr>";
else:
echo "<td><center><img src=\"image\\nodel.gif\" border=0></a></td><tr>";
endif;
endwhile;
echo "</table>";
- zobrazení přiřazených PC k příslušnému záznamu

ODBC_Close($spojeni); ?>
</html>
- uzavření spojení do databáze a ukončení HTML stránky
```

## 9.2 Skript pro tisk do PDF

```
<?
if (!isset($jev_id_old)): $jev_id_old = "0";endif;
if (!isset($topol)): $topol = "";endif;
if (!isset($shp_nazev)): $shp_nazev = "";endif;
if (!isset($pocetshp)): $pocetshp = 0;endif;
- první inicializace proměnných

define('FPDF_FONTPATH','./font/');
require('./fpdf.php');

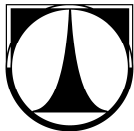
class PDF extends FPDF
{
function Header()
{
//Záhlaví stránky
$this->SetY(5);
$this->SetFont('times','',14);
$this->SetTextColor(255,0,0);
$this->Cell(0,10,'Tisková sestava - Software',0,1,'C');
}

function Footer()
{
//Zápatí
$this->SetY(-15);
$this->SetFont('Arial','',8);
$this->SetTextColor(128);
$this->Cell(0,10,'Stránka '.$this->PageNo(),0,0,'C');
}
}
- definice PDF a podoby stránky

$pravo = "admin";

$pdf=new PDF();
$pdf->Open();
$pdf->AddFont('arialce','', 'arial.php');
$pdf->AddPage();
$spojeni = ODBC_Connect("DRIVER={SQL
SERVER};SERVER=sde;DATABASE=evidencepc","evidence","heslo1234");
$vysledek = ODBC_Exec($spojeni, "SELECT * FROM aplikace where ID=$ID ORDER BY ID" );
$vysledek3 = ODBC_Exec($spojeni, "SELECT * FROM apl_upd where ID_P=$ID ORDER BY ID_U" );
- spojení do databáze a výběr dat dle vstupujících proměnných

while (ODBC_Fetch_Row($vysledek)):
$p_cislo = ODBC_result($vysledek, "ID");
$inv_cislo = ODBC_result($vysledek, "inv_cislo");
$nazev = ODBC_result($vysledek, "nazev");
$verze_p = ODBC_result($vysledek, "verze_p");
$verze_a = ODBC_result($vysledek, "verze_a");
$autor = ODBC_result($vysledek, "autor");
$dodavatel = ODBC_result($vysledek, "dodavatel");
$zavazky = ODBC_result($vysledek, "zavazky");
$typ_lic = ODBC_result($vysledek, "typ_lic");
$typ_prog = ODBC_result($vysledek, "typ_prog");
```



```
$poc_lic = ODBC_result($vysledek, "poc_lic");  
$spravce = ODBC_result($vysledek, "spravce");  
$pozn = ODBC_result($vysledek, "pozn");  
$popis = ODBC_result($vysledek, "popis");  
- naplnění proměnných daty z databáze
```

```
$pdf->SetFont('arialce','',10);  
$pdf->Cell(30,5,"Program:", "B",1,"c");  
$pdf->SetFont('arialce','',6);  
$pdf->Cell(20,5,"Pořadové číslo", "TBLR",0,"c");  
$pdf->SetFont('arialce','',8);  
$pdf->Cell(50,5,$p_cislo, "TBLR",0,"c");  
$pdf->SetFont('arialce','',6);  
$pdf->Cell(20,5,"Inventární číslo", "TBLR",0,"c");  
$pdf->SetFont('arialce','',8);  
$pdf->Cell(50,5,$inv_cislo, "TBLR",1,"c");  
$pdf->SetFont('arialce','',6);  
$pdf->Cell(20,5,"Název programu", "TBLR",0,"c");  
$pdf->SetFont('arialce','',8);  
$pdf->Cell(120,5,$nazev, "TBLR",1,"c");  
$pdf->SetFont('arialce','',6);  
$pdf->Cell(20,5,"Verze počáteční", "BL",0,"c");  
$pdf->SetFont('arialce','',8);  
$pdf->Cell(50,5,$verze_p, "BL",0,"c");  
$pdf->SetFont('arialce','',6);  
$pdf->Cell(20,5,"Verze aktuální", "BLR",0,"c");  
$pdf->SetFont('arialce','',8);  
$pdf->Cell(50,5,$verze_a, "BLR",1,"c");  
$pdf->SetFont('arialce','',6);  
$pdf->Cell(20,5,"Autor programu", "TBLR",0,"c");  
$pdf->SetFont('arialce','',8);  
$pdf->Cell(120,5,$autor, "TBLR",1,"c");  
$pdf->SetFont('arialce','',6);  
$pdf->Cell(20,5,"Dodavatel", "TBLR",0,"c");  
$pdf->SetFont('arialce','',8);  
$pdf->Cell(120,5,$dodavatel, "TBLR",1,"c");  
$pdf->SetFont('arialce','',6);  
$pdf->Cell(20,5,"Smluvní závazky", "TBLR",0,"c");  
$pdf->SetFont('arialce','',8);  
$pdf->Cell(120,5,$zavazky, "TBLR",1,"c");  
$pdf->SetFont('arialce','',6);  
$pdf->Cell(20,5,"Typ licence", "BL",0,"c");  
$pdf->SetFont('arialce','',8);  
$pdf->Cell(50,5,$typ_lic, "BLR",0,"c");  
$pdf->SetFont('arialce','',6);  
$pdf->Cell(20,5,"Typ programu", "BLR",0,"c");  
$pdf->SetFont('arialce','',8);  
$pdf->Cell(50,5,$typ_prog, "BLR",1,"c");  
$pdf->SetFont('arialce','',6);  
$pdf->Cell(20,5,"Počet licencí", "BL",0,"c");  
$pdf->SetFont('arialce','',8);  
$pdf->Cell(50,5,$poc_lic, "BL",0,"c");  
$pdf->SetFont('arialce','',6);  
$pdf->Cell(20,5,"Správce", "BLR",0,"c");  
$pdf->SetFont('arialce','',8);  
$pdf->Cell(50,5,$spravce, "BLR",1,"c");  
$pdf->SetFont('arialce','',6);  
$pdf->Cell(20,5,"Poznámka", "TBLR",0,"c");  
$pdf->SetFont('arialce','',8);  
$pdf->Cell(120,5,$pozn, "TBLR",1,"c");  
$pdf->SetFont('arialce','',6);  
$pdf->Cell(20,5,"Popis programu", "TBLR",0,"c");  
$pdf->SetFont('arialce','',8);  
$pdf->Cell(120,5,$popis, "TBLR",1,"c");  
- vlastní tabulka aplikace
```

```
$pdf->SetFillColor(128,128,128);  
$pdf->Cell(120,5,"", "",1,"c");  
$pdf->SetFont('arialce','',10);  
$pdf->Cell(30,5,"Update a upgrade:", "",1,"c");  
$pdf->SetFont('arialce','',8);  
$pdf->Cell(28,4,"Faktura", "BLT",0,"c");  
$pdf->Cell(28,4,"Verze", "BLT",0,"c");  
$pdf->Cell(28,4,"Licence", "BLT",0,"c");  
$pdf->Cell(38,4,"Poznámka", "BLT",0,"c");  
$pdf->Cell(18,4,"Datum", "BLRT",1,"c");  
$pdf->SetFont('arialce','',6);
```



```
while (ODBC_Fetch_Row($vysledek3)):
    $faktura = ODBC_result($vysledek3, "faktura");
    $verze = ODBC_result($vysledek3, "verze");
    $poc_lic = ODBC_result($vysledek3, "poc_lic");
    $pozn = ODBC_result($vysledek3, "pozn");
    $datum = ODBC_result($vysledek3, "datum");
    $pdf->Cell(28,3,$faktura,"BLT",0,"c");
    $pdf->Cell(28,3,$verze,"BLT",0,"c");
    $pdf->Cell(28,3,$poc_lic,"BLT",0,"c");
    $pdf->Cell(38,3,$pozn,"BLT",0,"c");
    $pdf->Cell(18,3,$datum,"BLRT",1,"c");
endwhile;
    - zobrazení upgradu pro příslušný záznam

$pdf->Cell(120,5,"", "",1,"c");
$pdf->SetFont('arialce','',10);
$pdf->Cell(30,5,"Přiřazená PC:", "",1,"c");
$pdf->SetFont('arialce','',8);
$pdf->Cell(28,4,"ID počítače", "BLT",0,"c");
$pdf->Cell(28,4,"Jméno PC", "BLT",0,"c");
$pdf->Cell(28,4,"IP adresa", "BLTR",1,"c");
$pdf->SetFont('arialce','',6);
$prir_pc = ODBC_Exec($spojeni, "SELECT * FROM pc_to_soft where id_soft = $ID ORDER by id_pc");
while (ODBC_Fetch_Row($prir_pc)):
    $n_pc = ODBC_result($prir_pc, "ID_PC");
    $nalez_pc = ODBC_Exec($spojeni, "SELECT * FROM pocitace where id_pc = $n_pc ORDER by
id_pc");
    while (ODBC_Fetch_Row($nalez_pc)):
        $id_pc = ODBC_result($nalez_pc, "id_pc");
        $jmeno = ODBC_result($nalez_pc, "jmeno");
        $ip = ODBC_result($nalez_pc, "ip");
        $pdf->Cell(28,3,$id_pc,"BLT",0,"c");
        $pdf->Cell(28,3,$jmeno,"BLT",0,"c");
        $pdf->Cell(28,3,$ip,"BLTR",1,"c");
    endwhile;
endwhile;
    - zobrazení přiřazených počítačů

$pdf->Output();
endwhile;
?>
    - vlastní generování do PDF a ukončení skriptu
```

**Seznam souborů na CD:**

**FONT** – adresář s fonty pro FPDF knihovnu

**IMAGE** – adresář s obrázky

**PDF** – adresář s FPDF knihovnou

**edit\_upd.php** – editace upgradu

**fpdf.css** – CSS soubor pro FPDF knihovnu

**fpdf.php** – FPDF knihovna

**index.php** – přihlašovací dialog

**index\_pc.php** – výpis PC

**index\_propoj.php** – výpis propojení PC x soft

**index\_soft.php** – výpis softwaru

**index\_tisk.php** – tisková sestava

**init.php** – inicializační soubor s připojením k DB

**maz\_pc.php** – výmaz PC

**maz\_upd.php** – výmaz update

**maz\_uziv2.php** – výmaz uživatele

**maz\_zaznamy.php** – výmaz software

**pridat\_pc.php** – přidání PC

**pridat\_pc\_zapis.php** – zápis přidaneho PC

**pridej\_upgr.php** – přidání upgradu k software

**pridej\_upgr\_zapis.php** – zápis přidaneho upgrade

**prirad\_pc.php** – přiřazení PC k soft

**styly.css** – CSS soubor

**tisk\_zaznamy.php** – export 1 stránky s popisem SW do PDF

**tiskpc.php** – export do PDF dle PC

**tisksoft.php** – export do PDF dle SW

**ukaz\_uzivatel.php** – evidence přihlášení uživatelů

**uprav\_pc.php** – úprava PC

**uprav\_pc\_zapis.php** – zápis upraveného PC

**uprav\_uziv.php** – správa uživatelů

**uprav\_uziv2.php** – změna hesla

**uprav\_zazn.php** – úprava softwaru

**uprav\_zazn\_zapis.php** – zápis úpravy software

**vloz\_uziv2.php** – vložení uživatele

**vloz\_zazn.php** – vložení software

**vloz\_zazn\_zapis.php** – zápis vloženého software

**skripty.sql** – soubor se skripty pro vytvoření tabulek, uživatelů a prvotní nahrání dat



## 10. Závěr

Prvotním impulsem pro vytvoření aplikace byl požadavek na evidenci softwarového vybavení od mých spolupracovníků. Po dohodě jsme zvolili formu intra/internetové aplikace, která dovolí správu této evidence z více míst, popř. nahlížení do ní mimo pracoviště.

Aplikace byla prvotně určena pouze pro evidenci programů, vývojem byla rozšířena o evidenci PC a evidenci umístění softwaru na určitém PC. V konečné fázi byla implementována uživatelská práva (čtenář, editor, administrátor) a propojení s tiskovou knihovnou FPDF pro tvorbu jednoduchých tiskových sestav.

Jak již bylo zmíněno, aplikace má implementovány 3 typy uživatelských rolí, od nejjednodušší role „čtenář“ (právo pouze zobrazovat záznamy bez možnosti přidávání, editace či mazání), přes „editora“ (právo přidávat a editovat, nemožnost výmazu) až po roli „administrátor“ (právo na úplnou správu databáze).

Na žádost uživatelů byla aplikace rozšířena o tiskovou knihovnu FPDF umožňující tvorbu jednoduchých tiskových sestav, ať již pro jednotlivý záznam, či celou evidenci do formátu PDF.

Aplikace je v současné podobě velmi využívána, kolegové si ji velmi oblíbili, možná i z důvodu, že byla vyvíjena dle jejich specifikací a mnohokrát byl její tvar upravován dle jejich připomínek.

Jsem si vědom, že se aplikace nemůže srovnávat s profesionálními evidencemi, hlavně v oblasti vizuální (tedy grafické) podoby, ale při její tvorbě byl kladen důraz více na funkčnost a jednoduchost než líbivý vzhled.

Vývoj aplikace nebyl zatím dokončen, do budoucna se plánuje zlepšení vizuální podoby, aby více vyhovovala současným trendům, dále je v uvažování rozšíření o dálkovou správu PC, resp. zjišťování softwaru na konkrétním PC přes LAN.



## Použité zdroje:

### Literatura:

Kosek, J. 1998. PHP - tvorba interaktivních internetových aplikací, Grada Publishing, Praha, 492 st., ISBN 80-7169-373-1

Whalen E, Garcia M. Microsoft SQL Server 2005 – Administrator's Companion, Microsoft Press, Redmint, 1186 st., ISBN 0-7356-2198-5

### Použité internetové stránky:

[1]

<http://www.zive.cz/Clanky/Vite-co-je-SQL-Ne-Nevadi---dnes-zaciname/sc-3-a-4320/default.aspx>

[2]

<http://www.microsoft.com/cze/windowsserversystem/sql/prodinfo/overview/default.mspix>

[3]

překlad z <http://www.microsoft.com/sqlserver/2008/en/us/express.aspx>

[4]

<http://interval.cz/clanky/php-zakladni-informace/>

[5]

<http://www.webtvorba.cz/php/uvod-do-php.html>

[6]

<http://www.fpdf.cz/co-je-fpdf/>

### Další internetové stránky zabývající se vybranou problematikou:

#### PHP

Oficiální stránky o PHP – [www.php.net](http://www.php.net)

JakPsátWeb - <http://www.jakpsatweb.cz/php/jak-zacit.html>

Wikipedia - <http://cs.wikipedia.org/wiki/PHP>

#### FPDF

Oficiální stránky - <http://www.fpdf.org/>

České stránky - <http://www.fpdf.cz/>

#### MSSQL

[http://en.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://en.wikipedia.org/wiki/Microsoft_SQL_Server)